Platform

Security Functional Requirements

PlatformSecurityFRDv0.52

Revision History

Date	Version	Description	Author
2/19/01	0.1	Initial Version	Jayashree Dorairaj
2/25/01	0.11	Based on Edwin Desouza's feedback, expanded sections for 1. LDAP synchronization, Encryption etc. 2. Added a section on wireless Security.	Jayashree Dorairaj
3/5/01	0.12	Made changes based on E. Desouza's & Sachi Begur's feedback. 1. Added a section on SSO – Types of Apps, Jamcracker Security Philosophy. 2. Added retrieval of passwords.	Jayashree Dorairaj
3/9/01	0.2	Made the following changes based on Edwin, Anand, Gil's feedback. 1. Reworded and re-arranged sections. Removed Targeted Audience (per Anand's feedback). 2. Moved Glossary of terms to last section. 3. Removed section on Anonymous Logins, Concurrent Sessions, SSO types of applications. 4. Added Section on Password Policies. 5. Added advantages of using standards/protocols.	Jayashree Dorairaj
3/16/01	0.3	Made the following changes based on Edwin, Anand, Sachi, Raghu's feedback 1. Changed bulleted items to sub-headings on Password Policies. 2. Added more details on Delegated Administration. 3. Moved Roes and Privileges to a separate section. 4. Added auditing requirement for LDAP-LDAP synchronization. 5. Reorganized section on security Requirements. 6. Renamed LDAP synchronization to Meta Directory Synchronization. 7. Added section on SAS-70.	Jayashree Dorairaj
3/22/01	0.4	Made the following changes based on Gil's feedback. 1. Added, "Targeted audience". 2. Removed SAS-70. 3. Removed types, strength and data fields for encryption (Gil & Raghu's feedback) 4. Added more diagrams, details for Delegated administration.	Jayashree Dorairaj
3/28/01	0.41	Made the following changes based on Edwin's Feedback. 1. Removed Glossary of terms. 2. Added more details on Authorization – Policies, Domains. Groups etc.	Jayashree Dorairaj
4/2/01	0.42	Made the following changes: 1. Split Secure SSO – Platform as Spoke to Platform and ASP as spoke. 2. Added user time-out to Session Management.	Jayashree Dorairaj

		3. Added sub-headings to ASP, Company, User and Syndicator Security requirements so it can be tracked through features matrix.	
4/9/01	0.43	Made the following changes based on Sachi, Edwin, Gil, Sean Murphy's feedback 1. Added more details on required rights model – Authorization. 2. Added Security Requirements for outsourced partners. 3. Multiple Concurrent Sessions. 4. Added section on Username and Company Name Changes.	Jayashree Dorairaj
4/18/01	0.44	Added DSML to list of standards.	Jayashree Dorairaj
4/20/01	0.5	Made the following changes based on Vinay Singla, Daniel Durocher's feedback. 1. Added Secure Email, Secure FTP, JCE/JCA, JSSE to standards and protocols section. 2. Added types, strength and data fields for encryption. 3. Preferred Strength of SSL, PKI, Preferred algorithms 4. Added Company name change, username changes for logging/auditing. 5. Added comments on minimum standards for JC password policies, floor value for password history etc.	Jayashree Dorairaj
6/11/01	0.51	Made the following changes based on SSO design proposal from PM (myself), Architecture (Gil) and Edwin. 1. Expanded section on First-time passwords 2. Expanded section on Reset Passwords 3. Expanded section on SSO to Non-Affiliated ASPs. 4. Requirements for Third-Party CSR Roles	Jayashree Dorairaj
7/3/01	0.52	Made the following changes based on discussion with JSC 1. Expanded requirements on account locking, account disabling, session management – concurrent logins and permanent cookies	Jayashree Dorairaj

Contents

1. Introduction

1.1. What is covered?

This document covers the security requirements for the Jamcracker workspace. This includes Customers, Partners, and Syndicators from a platform perspective. It covers requirements in the areas of authentication, authorization, session management, LDAP synchronization and encryption. All the requirements arise from the fundamental need to fulfill the 5 security goals – Confidentiality, Integrity, Availability, Accountability and Assurance.

The first section of the FRD (section 4) discusses the recommended technology and standards.

The second part of the FRD (sections 5 through 14) covers requirements that are generic in nature and applicable for all types of players in the JC Ecosystem – Customer, Partner, Syndicator and User. Sections 5 through 14 are as follows: -

Section 5 - Authentication
Section 6 – Authorization
Section 7 - Encryption
Section 8 – Secure Single Sign-On
Section 9 – Session Management
Section 10 – Roles and Privileges
Section 11 – Delegated Administration
Section 12 – Auditing & Logging
Section 13 – Meta Directory Synchronization
Section 14 – Password Policies

The third part of the FRD (sections 15 through 19) covers security requirements for the various types of players in the Jamcracker ecosystem. It includes:

Section 15 – Syndicator Security Requirements
Section 16 – Company Security Requirements
Section 17 – User Security Requirements
Section 18 – ASP Security Requirements
Section 19 – Security Requirements For Outsourced Partners

This section contains references to the generic security requirements discussed in the second part of the FRD.

Section 20 contains a section on Changing Usernames and Company Names.
Section 21 contains a list of configurable items.
Section 22 on wireless security will be expanded later.

It is assumed that reader is familiar to a certain degree with the following terms – Authentication, Authorization, Delegated Administration, etc.

1.2. What is not covered?

This document does not attempt to cover security requirements such as

- Security policies and procedures applicable to both Jamcracker and the ASPs
- Legal implications for ASP partners such as bankruptcy etc,
- Handling of security incidents such as ICMP, DOS attacks etc.
- Network Security requirements such as establishing secure tunnels, VPN etc

1.3. General Note Across the FRD

Intent of this FRD is to focus on the “What” part of a requirement rather than the “how”.

A pre-requisite for this FRD is the Roles & Privileges FRD.

The following terms are used across the FRD:
JC – refers to Jamcracker.
Terms Customer and Company is used interchangeably.

1.4. References

Roles and Privileges FRD v 0.5– Anand Ranthidevan
User Management FRD v0.5 – Anand Ranthidevan
Provisioning and services Management FRD v0.5 – Anand Ranthidevan
Billing FRD v0.5 – Kiran Kamtekar
Jamcracker Platform Security White Paper – Gil Pilz
LDAP Synchronization v0.1 – Raghu Chakravarthi
Enterprise Security Architecture in a multi-domain networked infrastructure v0.1 – Sachi Begur

1.5. Targeted Audience

Engineering, Architecture, Product Management Teams, Security Vendors.

2. Jamcracker’s Security Philosophy

Jamcracker’s overall security philosophy is

“To enable an organization to meet all mission/business objectives, while ensuring that system implementations demonstrate due care and consideration of risks to the organization and its customers. This purpose is achieved by accomplishing the five security goals described below.”

- Availability
- Integrity
- Confidentiality
- Accountability
- Assurance

2.1. Availability

Availability ensures that systems are accessible and usable when needed. It implies the additional requirement that systems need to be protected against intentional or accidental attempts to:

- Perform unauthorized deletion of data.
- Cause a denial of service or data.
- Use system for unauthorized purposes.

2.2. Confidentiality

Confidentiality refers to prevention of unauthorized or undesirable disclosure of information.

2.3. Integrity

Integrity includes:

- Data integrity (the property that data has not been altered in an unauthorized manner while in storage, during processing, or while in transit).
- System integrity (the quality that a system has when it performs its intended function in an unimpaired manner, free from unauthorized manipulation).

2.4. Accountability

Accountability associates actions with responsible entities. It indirectly supports non-repudiation, deterrence, fault isolation, intrusion detection and prevention, after-action recovery and legal action.

2.5. Assurance

Assurance ensures that other 4 goals are sufficiently met
This includes:

- Functionality that performs correctly
- Sufficient protection against unintentional errors (by users or software)
- Sufficient resistance to intentional penetration or by-pass

Assurance is essential; without it the other goals are not met. Assurance is a continuum; the amount of assurance needed varies between systems.

3. Security in the Jamcracker Ecosystem

The Jamcracker ecosystem is a network of trusted partners offering web services to Jamcracker customers. We use the term 'ecosystem' to convey the organic, interdependent nature of the relationships between Jamcracker and these partners.

The following depicts a pictorial representation of a Jamcracker ecosystem.

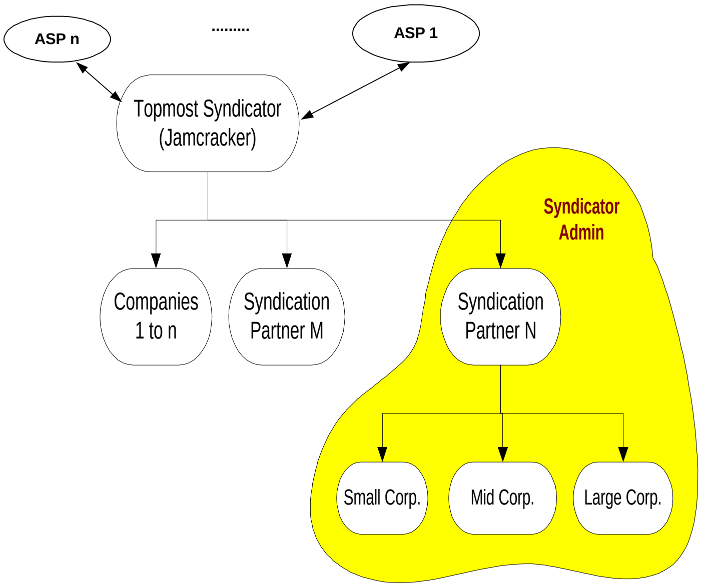


Figure 1: Syndicator Model

ASP Partners – They are the building blocks of Jamcracker's ecosystem. Jamcracker's commitment to meet the security goals of Availability, Confidentiality, Integrity, Accountability and Assurance extend beyond its operational boundaries to include its partners as well. Jamcracker has a number of integration points with its partners, including Single Sign-On, provisioning and de-provisioning. During provisioning and de-provisioning, data exchange takes place between JC and ASP partner to activate and de-activate the service for the end-users.

Security requirements of data privacy, integrity and authentication must be maintained during provisioning and de-provisioning.

Syndicators - The entire model is based on the idea of efficiently and reliably providing web services. Syndication is a by-product of this goal. It is based on the assumption that the Jamcracker Workspace will be resold to other “Syndication Partners”– who will customize it, configure it, and choose to either further resell it or offer it directly to end Customers. In such a hierarchical model, Jamcracker is the topmost Syndicator.

Customers – Customer always accesses the platform through a syndicator. (Jamcracker being the topmost syndicator.)

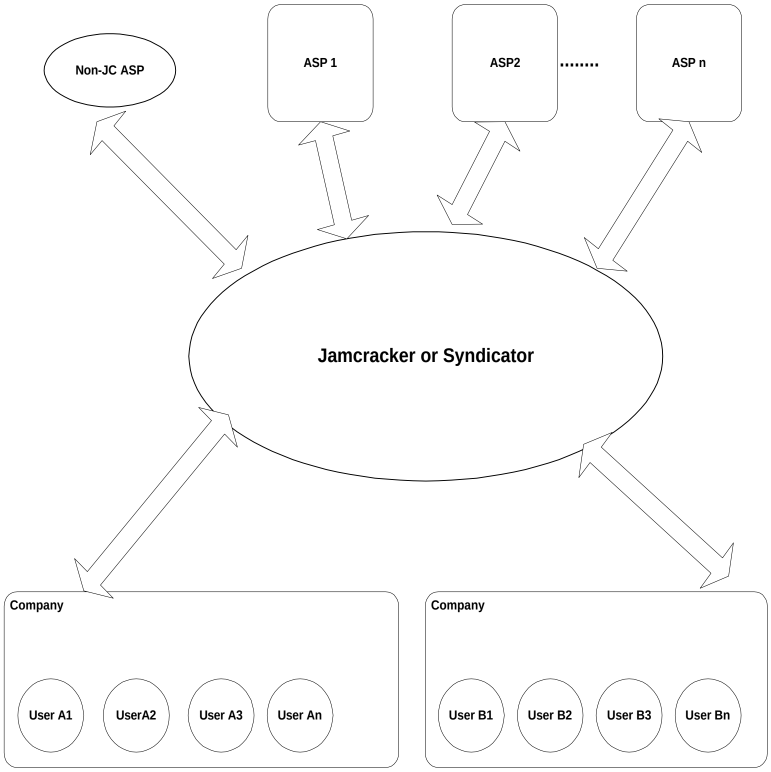


Figure 2 – Ecosystem Players

The security vision is to provide **end-to-end security**. From the time

- A user fires up her browser.
- Access the JC workspace (Authentication).
- Sees personalized content that she is supposed to see (Authorization).
- Clicks on different links to access different ASP services without having to re-login (SSO).
- Establishes a secure session (Session Management & Encryption) so no one eavesdrops.
- Logs out of the Workspace.

He or she should trust JC in managing critical data and systems needed to fulfill his or her Company's business objectives.

From a security perspective, JC must be able to do the following: (More details on each of these areas are in the sections to follow.)

- Authentication - Support Multiple Forms (basic, strong), Authentication with no password store and travel, Multiple Levels, Alternate methods of authentication.
- Authorization - Support Authorization policies based on Roles and Privileges and Centralized Authorization Model.
- Password Policies – Allow for Companies to realize their password policies and support functionality.
- Secure SSO - Support SSO (Single Sign-On) functionality securely to various types of applications.
- Session Management - Support global logout, source timeouts and destination timeouts. Also be able to track users sessions from login to logout.

- Delegated Administration - Support delegated administration so that Roles & Privilege management activities can be securely delegated to the Syndicators or the Companies themselves.
- Auditing and Logging - Log all administrator related activities, resource usages, intrusion detection and audit any events specified by Company.
- Encryption - Encrypt all customer sensitive data and all sensitive communication.
- Meta Directory Synchronization - Support security requirements of data privacy, integrity, non-repudiation and audit trail availability during directory synchronization.

Hence, JC should provide a security infrastructure that satisfies the above goal.

4. Security – Adherence to Standards & Protocols

4.1. Overview

This section covers various standards and protocols that are recommended. The benefits of adhering to open standards are ease of integration to a range of applications that are heterogeneous in nature rather than dealing with proprietary protocols and data formats.

4.2. HTTPS

Protocol for accessing a secure Web server. It is the HTTP protocol built on top of the SSL protocol. It provides data integrity and user-data confidentiality.

4.3. SSL

SSL stands for Secure Socket Layer. It provides strong encryption and authentication to protect data going over the Internet. SSL solves the problem of transport layer integrity and confidentiality.

4.4. PKI

PKI – Public Key Infrastructure provides a distributed security infrastructure to support strong authentication, authorization and non-repudiation for users and servers based on digital certificates.

4.5. LDAP

LDAP stands for Lightweight Directory Access Protocol. It derives its origin from X.500. There is a strong trend in most enterprises towards the adoption of LDAP as their enterprise directory infrastructure. This is because it provides the critical features of a hierarchical directory, and is optimized for the access patterns that an access management system would have (in particular, a very high number of Read Operations).

4.6. XML

A markup language that, like HTML, is derived from the Standard Generalized Markup Language but is far more flexible. XML security standards provide the benefit of unified format for any-to-any exchange of security data.

4.7. SOAP

SOAP – Simple Object Access Protocol is an XML/HTTP-based protocol for accessing services, objects and servers in a platform-independent manner.

4.8. SAML

The Security Assertion Markup Language (SAML) is a set of XML schemas and interfaces for security services. SAML provides a standard description of authentication and authorization as XML response pairs.

4.9. DSML

DSML – Directory Services Markup Language is a standard way of representing directory services in XML.

4.10. JAAS

JAAS stands for Java Authentication and Authorization services, a Java package that enables services to authenticate and authorize users. It is characterized by its user-centric access control rather than code-centric.

4.11. JCE/JCA

JCE – Java Cryptographic extension is a package that provides a framework and implementations for encryption, key generation and key agreement, and Message Authentication Code (MAC) algorithms.

4.12. JSSE

JSSE – Java Secure Socket Extension. A Java package that enables secure internet communications. It implements a Java version of SSL (Secure Sockets Layer) and TLS (Transport Layer Security) protocols and includes functionality for data encryption, server authentication, message integrity, and optional client authentication.

4.13. VPN

VPN – Virtual Private Network. VPN securely connects the components of one public network to another public network, accomplishing this through the use of encryption and other security tools that allow the user to tunnel through the Internet or another public network in a manner that provides security and features formerly available only in private networks.

4.14. Secure FTP

Secure FTP is a client package that allows for a secure connection to be made to an FTP daemon. It supports encryption through SSL.

4.15. Secure Email (S/MIME, PGP)

S/MIME is a specification for secure electronic mail. S/MIME stands for Secure/Multipurpose Internet Mail Extensions and was designed to add security to e-mail messages in MIME format. S/MIME incorporates digital certificates, the online passports that authenticate the identity of an email's sender and receiver, verify message integrity, and ensures the privacy of the message's content.

PGP® or Pretty Good Privacy® is a powerful cryptographic product family that enables people to securely exchange messages, and to secure files, disk volumes and network connections with both *privacy* and *strong authentication*.

5. Authentication

Protection of sensitive resources is absolutely critical to any business. Businesses must feel confident that only valid users can access their resources and that any unauthorized or suspicious user activity must be dealt with immediately. However, security on the site must not be so cumbersome or intrusive to manage that it becomes difficult to do one's job. Hence, any solution must offer a variety of ways of securing sensitive resources.

The security infrastructure must be capable of supporting a range of authentication methods. Furthermore, the ability to change from one method of authentication to another must be seamless to a customer. If at all possible, the changes should be limited to configuration rather than code.

5.1. Basic Authentication

A basic form of authentication using a company/username/password combination must be available.

5.2. Strong authentication

The infrastructure should support strong forms of authentication such as digital certificates, especially to sensitive applications and protected resources.

For instance, some customers would want to have PKI-digital (X.509) certificate based strong form of authentication for certain types of financial applications.

5.3. Multiple Levels of Authentication

This functionality must be available when certain sensitive or confidential applications require a greater level of security. For instance, certain modules within a financial application can only be viewed by a specific set of users and thus, require additional authentication. In that case, the security infrastructure should be capable of handling multiple levels of authentication

Another example is when all users could be authenticated first with a username/password combination over SSL, but when he or she accesses a protected resource, a certificate-based authentication could take place.

5.4. Alternate Methods of Authentication

Alternate methods of authentication must be available to provide administrative flexibility particularly when there is a need to gradually deploy certificates to large user populations.

For instance, some Companies would like to authenticate using certificates if one is installed. If one is not present, then the authentication would fallback to regular password. This would help in scenarios where a User would use certificates when they are at their office location. However, if he or she was traveling and wanted to use public terminals to access the workspace, a password-based authentication would take place

Another example would be, if a global Company wanted to use certificate-based authentication for all of its users in the U.S. and password based authentication for all of its international users.

5.5. Authentication With No Password Store At JC

The Security infrastructure must be capable of authenticating customers who do not wish to store their passwords at JC. For instance, passwords could come from the following sources.

5.5.1. Located in Customer Repository

Here the password comes from the Customer repository (typically LDAP-based) and the platform should be capable of authenticating against it.

5.5.2. Located in Physical Devices

In this case, customers carry Smart Cards or SecurID cards and would want the security infrastructure to authenticate against them.

5.5.3. Located in Human Body

Customers should be able to authenticate using biometrics such as fingerprints or voice or retina scan.

5.6. Authentication With No Password Transmission

Ideally, passwords should not be transmitted during the process of authentication. But, if passwords are transmitted over the network for some reason, they should be encrypted.

.

6. Authorization

Authorization is the process of determining who can access which resource and in what way. For instance, applying business Rules to "who gets to see what". This is sometimes referred to as "access control". Users of the JC workspace should only be allowed to see content that they've been authorized to see. Resources could be anything such as: files, web pages, XML messages, content modules, EJB Methods, tickets etc.

Authorization Model should be

- Policy Based
- Centralized

6.1. Policy based authorization

The security infrastructure should support authorization policies based on roles and privileges.
For more details on Roles and Privileges, refer to the Roles and Privileges FRD.

Policy describes which user actions are allowed and which are forbidden. It also describes actions which the system must itself perform, and when it must perform them. Policy is expressed using policy rules. Policy rules must be flexible enough so that it reflects actual business rules such as "Vice-Presidents can approve expenses up to $10,000".

Another example where policy based authorization can be used is - an HR application where HR Admins can access/modify most data for a user. An employee can access and modify some of his data. An employee can access other employee's non-secure data (work phone, email address, mailing address, etc.). The model should be flexible enough so that any changes in the organization could easily be accommodated – such as an employee's direct manager will also have the ability to set their salary.

Policies should be flexible enough that it can be applied to more than one technology domain. For instance, when a customer decides that they want to become a JC customer directly instead of being a Syndicated customer, the roles and privileges change must happen seamlessly without any need to reprogram.

6.2. Policy Domains

Policy rules should be applied at a domain level, where each domain is a collection of resources that are considered equivalent from a policy perspective. For instance, a Company could have many different groups each of which could be grouped into a domain. (.e.g.) Business Development, VPs, Boston Sales, Finance etc.

In the case of the HR application example - The abstract roles we could use to control access to EJB methods are "HR", "Owner", and "Other". "HR" is a fixed group of people but "Owner" and "Other" is mapped dynamically based on if a user is trying to access his or another user's data. A Servlet and/or a session bean control this outside the EJB.

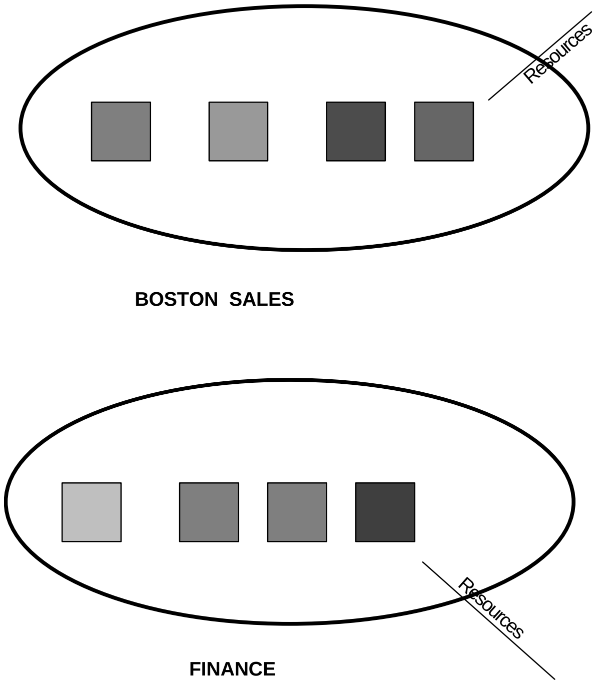


Figure 3: Policy Domains

6.3. Required Rights Model

Attached to each Policy Domain is an ACL (access control list) that defines the "effective" rights of any user or group of users within that Policy Domain. Policy Domains can be structured to relate to one another in a hierarchical fashion such that "child" Policy Domains inherits the policies of their "parent" Domains (and their parents parent etc.)

From a policy perspective, rules are applicable at a domain level. For instance for each of the EJB interfaces, one could define the rights required to execute each method like:

setSalary() - requires 'm' for manage and 'w' for write
getSalary() - requires 'm' for manage and 'r' for read
getOfficePhone() - requires 'r' for read.

In the required rights model, ACLs are never applied to resources directly, only to Policy Domains and that individual instances of resources are mapped to Policy Domains dynamically at run time. Instances of different EJBs/classes/resources can be placed in the same Policy Domain.

6.4. Hierarchical Policy Domains

Now a hierarchical tree of Policy Domains could be set up, where the root of this tree will simply be called "HR". The ACL on the "HR" domain states that the group "hr_admins" has rights "rwdm" and the group "other" (to which everyone belongs) has rights "r".

In a hierarchical policy domain lower level policy domains inherit policies from a higher-level policy domain (unless they are specifically overridden).

For each employee a unique Policy Domain could be constructed using some easy-to-construct mapping for the name (concatenate fist initial, last name, and employee number). Each of these employee domains could be placed directly under the root "HR" domain. The ACL for each of these employee domains states that the employee whose domain it is has rights "rm".

The factory that constructs each EJB instance must tell the security service where to place the bean in the Policy Domain hierarchy. The bean for each employee must be placed into the domain that corresponds to that employee.

To determine whether a user has access, the security system simply has to figure out the Policy Domain of the instance that the user is attempting to access. From there it can obtain the ACL and, based on the user's user/group/role attributes determine his or her "effective rights". To complete the picture the security service simply has to compare the user's effective rights against the required rights of the interface and method that the user is attempting to invoke.

For instance, if a user who is a member of "hr_admins" wanted to call setSalary() on a given EJB instance, he or she will be allowed to because the effective rights of the "hr_admins" group are "rwdm" for all the Policy Domains in the system (by inheritance) and the required rights for setSalary() are "wd". If the user wanted to view his or her own salary information, user would be allowed to because the effective rights for their own Policy Domain are "rm" and the required rights for getSalary() are "rm".

If user A wanted to view user B's salary he or she would *not* be allowed to because user A's record EJB lives in a different policy domain than user B. User A's effective rights in their policy domain are "r" (since he or she belongs to the group "other" and there is no other ACL entry that matches their attributes). Having the effective rights "r" with respect to their employee-record EJB does not allow them to execute getSalary() although it does allow to execute getOfficePhone().

In this model, the only component of this whole system that has to understand that certain beans "belong" to certain owners is the factory that constructs the beans and places them in the Policy Domain hierarchy. The rest of the security infrastructure just operates on "auto-pilot"; what domain is this instance in? What are the users effective rights? What are the required rights for this method?

In case, the system needs to incorporate additional changes such a manager can view his direct reports' salary information, all that is needed is to rearrange the Policy Domain hierarchy such that there is an intermediate level of domains defined for each manager. The ACL of each of these "manager level" domains would give that manager "rwdm" rights over that domain and all the domains beneath it. Then each of the "employee domains" can be placed under the domain of their manager.

The following picture depicts a hierarchical policy domain.

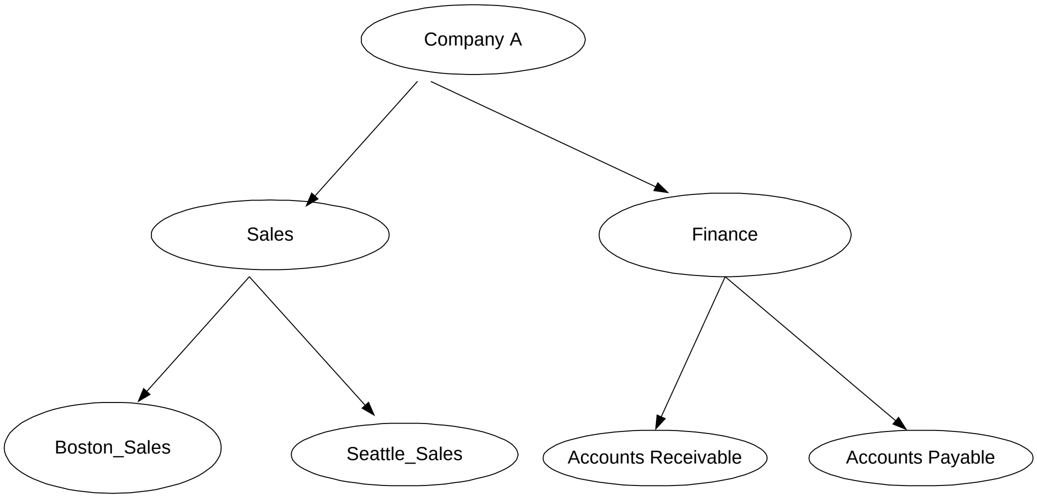
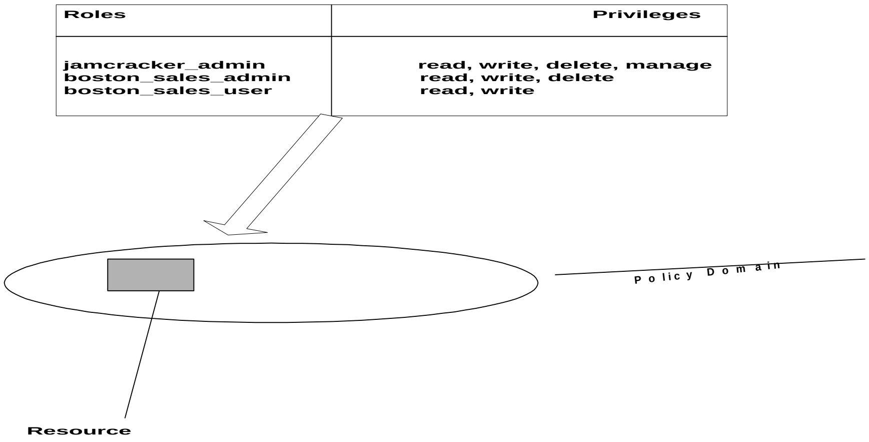


Figure 4: Hierarchical Policy Domain

6.5. Authorization Policies based on Roles and Privileges

Ideally, the security infrastructure will provide GUI tools to manage roles and privileges.
It should give an administrator the ability to not only define what a role or privilege is but also map it to protected resources. For instance, a Company administrator should be able to define which set of users can see what modules within a financial application. Authorization Policy determines the rights allowed to specific groups of users for a given policy domain.



Roles	Privileges
jamcracker_admin boston_sales_admin boston_sales_user	read, write, delete, manage read, write, delete read, write

Figure 5: Authorization Policy

6.6. Delegated Policy administration

Hierarchical Policy Domains are particularly well suited for implementing Delegated Administration. By organizing the Policy Domains in a manner that matches the companies internal organization and placing the appropriate policies in the appropriate places in the hierarchy, a natural delegation is enforced. Companies should be allowed to administer policies within their own domains in a delegated fashion. For instance, Companies should be able to establish their individual hierarchies responsible for managing specific resources and users, and allow sub-administration roles to possess limited administration powers, like allowing help desk personnel to reset a password.

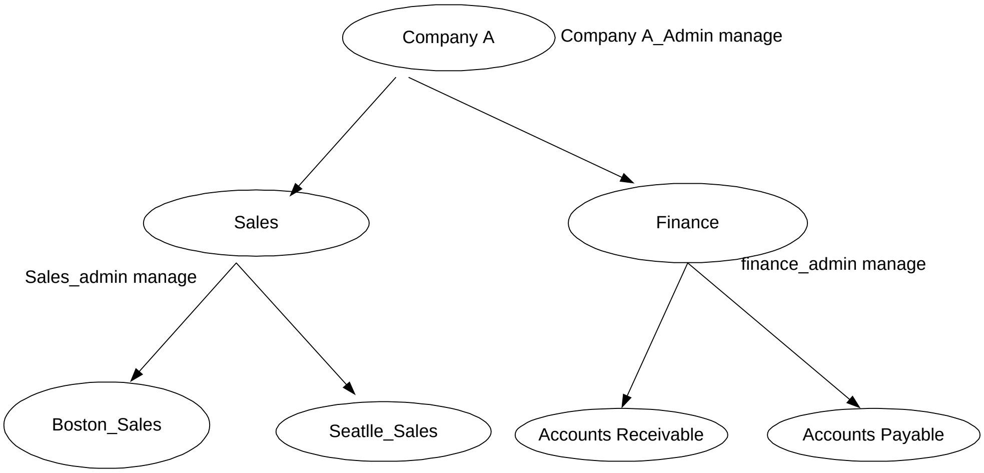


Figure 7: Delegated Policy Administration

6.7. Centralized Authorization

Security infrastructure should be capable of supporting a centralized authorization model. Centralized implies that all policies must be defined by JC or Syndicator administrator and managed centrally. The security policies should be consistent and easy to administer.

6.8. Centralized Authorization – Platform

Here authorization does not happen in the individual subsystems such as webserver, appserver, Workflow, Database etc. Instead the authorization logic is defined and managed centrally across all subsystems.

6.9. Centralized Authorization - ASPs

JC should not only be able to create roles and privileges for the JC workspace, but also capture what role or privilege a user has for the various applications that JC provides. JC should be able to map the business rules for its customers and translate them into authorization policies that ASP can understand. Although at this point in time, it is hard to imagine that ASP's would leverage runtime authorization logic from a central policy server hosted at Jamcracker.

For instance, if a company buys Employease (HR Services) from JC, we should be able to define what role each user has within the Employease application as well as define what type of access we want our CSRs (Customer Service Rep) to have on customer's sensitive data.

Another example would be when JC should have additional access controls so that a JC help desk person cannot execute a banking transaction on behalf of a customer but allowed limited access so as to be able to troubleshoot.

From an ASP perspective, there should be a clear delineation of authorization policies for JC customers only and non-JC customers.

7. Encryption

The encryption requirements from a platform security standpoint are that all sensitive customer data must be encrypted. What is deemed sensitive may vary from one Company to another.

The system should provide the ability to

- Encrypt any piece of information that is deemed sensitive at any point in time.
- For JC to specify what pieces of data should be encrypted. For instance, customer list, revenues, sales forecast numbers, network topology diagrams, pricing models, engineering documentation and schematics, etc. Here are some examples of data which are sensitive

  Passwords
  Social Security Number
  Birth Date
  Salary
  Customer Lists
  Sales forecast
  Revenues
  Network Topology Diagrams

7.1. Encryption Of Customer Sensitive Data at JC

All customer sensitive data at JC should be encrypted.

7.2. Encryption Of Customer Sensitive Data at ASP

All sensitive customer data at the ASP must also be encrypted.

7.3. Encryption Of All Customer Sensitive Communication

All communications between the Customer & JC, JC & the ASP and the Customer and the ASP that carry sensitive customer data must be encrypted. The communication could be a front-end session or a back-end communication (example, sending customer data across to the ASP to activate their accounts. For example, all email attachments containing customer sensitive data will be encrypted using PGP.)

7.4. Type and Strength of Encryption

Sensitive data could be encrypted using algorithms such as symmetric (DES, 3-DES), asymmetric (PGP, PKI) or hashing (MD5) etc. Degree of sensitivity dictates the strength of encryption. For example, passwords could be encrypted using 128-bit one-way hash, Social Security numbers using 3-DES etc. Data with lesser degree of sensitivity could be encrypted using 40-bit encryption.

8. Secure Single Sign-On

One of the most common problems faced in IT environments today is remembering multiple passwords for accessing different applications. This results in high administrative costs and user frustration.

With Single Sign-On (SSO) the user would have to authenticate only once and the resulting authentication assertions are passed amongst various applications that he or she is authorized to access. One of the value propositions of Jamcracker is to provide SSO for all its services.

SSO must be secure so that authentication information is not snooped during transmission from JC to the services (or ASP's).

The following sections discuss the variants of SSO.

In all the scenarios,

1. Source website represents a platform (for example, JC workspace, syndicated workspace, etc.)
2. Destination website could be any ASP or internal application that participates in SSO.

8.1. Secure SSO – Types Of Applications

The following table summarizes a list of various applications that could be integrated to provide SSO.

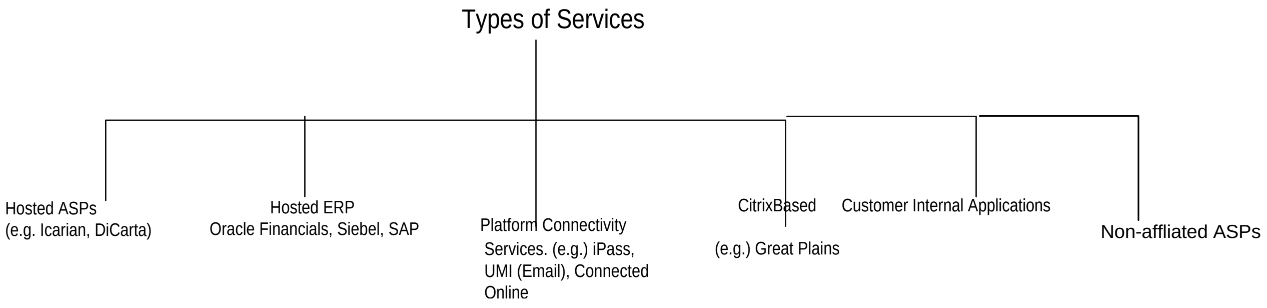


Figure 8: Type of Application*s*

Type Of Application	Example	Special Features
1. Hosted ASP	Icarian, DiCarta etc.	These are pure web-based applications.
2. Hosted ERP (html client based)	Siebel, Peoplesoft Oracle Financials	
4. Hosted ERP (Citrix-based)	Great Plains	
5. Customer Internal Application		SSO will be provided for web-enabled applications. For non-web enabled applications, a feasibility study will be done on a case-by-case basis.
6. Non-affiliated ASPs (NASP)	Salesforce.com, Get there.com, BlueMatrix – For Putnam Lovell	These ASPs are characterized by the fact that Jamcracker does not have any direct business relationship with them. The level of integration

		will be therefore very lite. The following are the key features: - 1. Jamcracker will not provision or de-provision the users at the NASP. 2. Jamcracker will not provide any support for application issues regarding the NASP except for SSO. 3. Jamcracker will not bulk-load authentication credentials for the NASPs as it is done today for affiliated ASPs. Instead, the platform will dynamically obtain the user credentials for authenticating with the NASP. 4. User may change password or authentication credentials at the NASP. However, the onus is on the user to correctly input those at Jamcracker platform in order for the SSO piece to work.
7. Connectivity based applications (client downloadable through the Platform)	iPass, UMI, Connected On-Line etc.	

8.2. Secure SSO with Platform as Hub

In this scenario, user logs into the JC or Syndicator platform, authenticates, clicks on any link that would take her to the destination website. Destination website should allow access to the user and show content based on what the user is authorized to see. User experience should be seamless in terms of accessing the destination website with just a single click. To access a second destination website, he or she would click on the corresponding link from the source website.

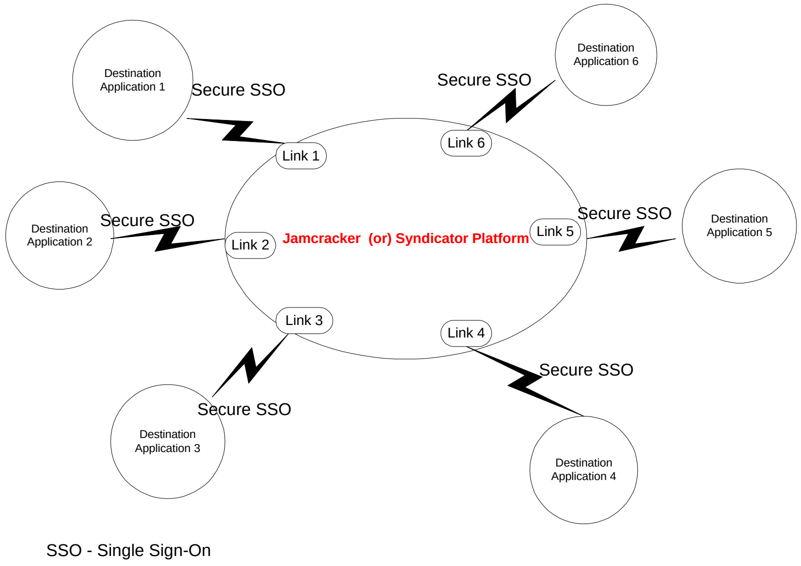

Destination Application 1
Secure SSO
Secure SSO
Destination Application 6
Link 1
Link 6
Destination Application 2
Secure SSO
Link 2
Jamcracker (or) Syndicator Platform
Link 5
Secure SSO
Destination Application 5
Link 3
Link 4
Secure SSO
Secure SSO
Destination Application 3
Destination Application 4
SSO - Single Sign-On

Figure 9: Secure SSO with Platform as Hub

8.3. Secure SSO with Platform as Spoke

In this case, user should be able to do the following:

b. Go to his or her portal.
c. Click on a link to JC portal.
d. Once on the JC portal, click on any of the ASP links that would seamlessly establish a session with the individual ASPs.

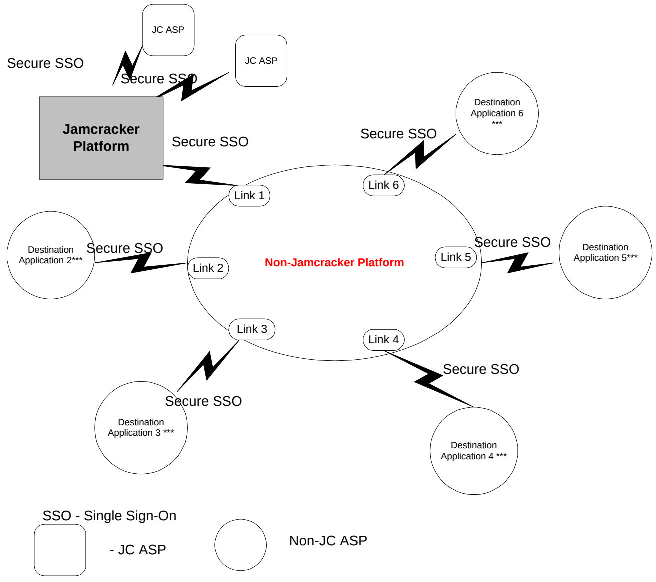


Figure 10: Secure SSO With JC Platform As Spoke

8.4. Secure SSO with ASP as Spoke

In this scenario, users should be able to do the following:
a. Go to his or her Company's portal.
b. Click on a JC ASP link
c. Seamlessly establish a session with the JC ASP. In the background, his or her portal co-operates with JC portal before firing the JC ASP link.

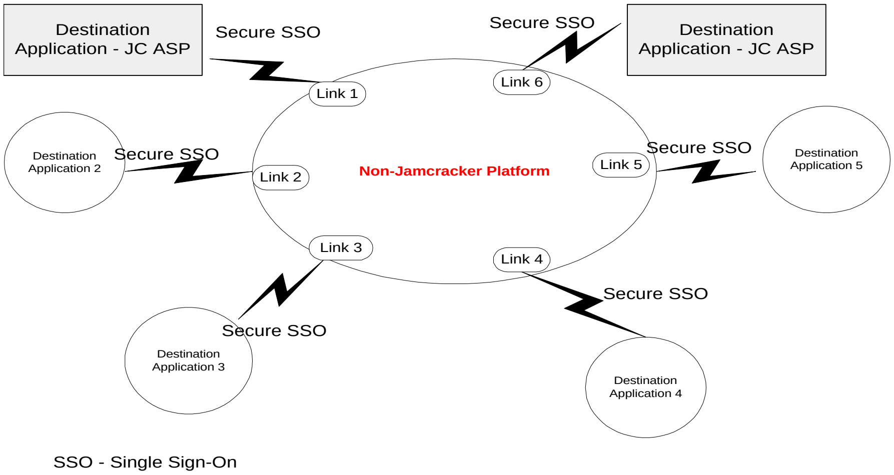

Destination Application - JC ASP
Secure SSO
Secure SSO
Destination Application - JC ASP
Link 1
Link 6
Destination Application 2
Secure SSO
Link 2
Non-Jamcracker Platform
Link 5
Secure SSO
Destination Application 5
Link 3
Link 4
Secure SSO
Secure SSO
Destination Application 3
Destination Application 4
SSO - Single Sign-On

Figure 10: Secure SSO with ASP as spoke

8.5. Secure SSO from ASP to ASP

In this scenario, the user would first authenticate with the source website, click on a link to a first destination website (ASP1) and establish a session. User should then be able to click on a link to ASP2 (second destination website) from ASP1. After clicking, a session should be established seamlessly with the second destination website. In all these cases, user does not have to come back to source website to access various destination websites.

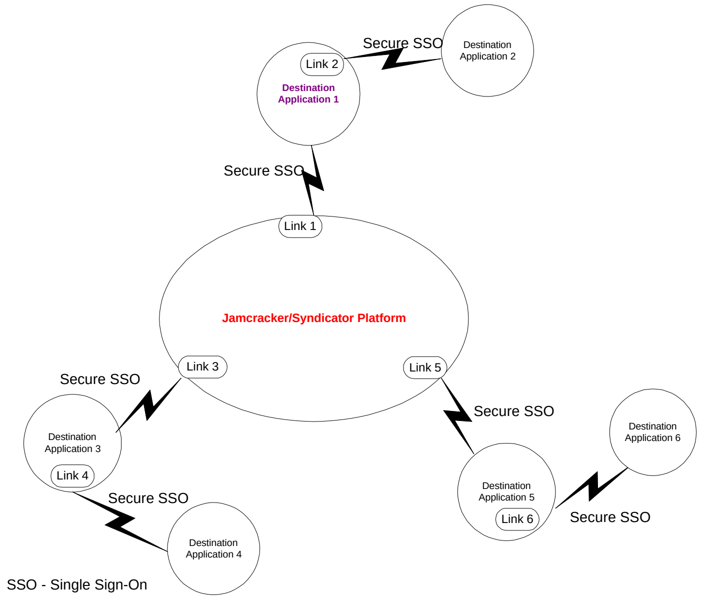

Secure SSO
Destination Application 2
Link 2
Destination Application 1
Secure SSO
Link 1
Jamcracker/Syndicator Platform
Secure SSO
Link 3
Link 5
Secure SSO
Destination Application 6
Destination Application 3
Link 4
Secure SSO
Destination Application 5
Link 6
Secure SSO
Destination Application 4
SSO - Single Sign-On

Figure 11: Secure SSO from ASP to ASP

8.6. Secure Virtual SSO

Here the user should be able to login to a hosted security service (could be JC or any service provider), authenticate and request a token or ticket with a predefined expiration time (say valid for 3 hours). Using the token, user should be able to click on the corresponding links for different ASPs from their desktops and establish a session seamlessly. The session with each of the ASPs last as long as the token is valid.

User should not have to go to JC workspace to click on the links. The ASPs that the user can access are all applications that participate in SSO.

8.7. Secure SSO For Connectivity Applications

Connectivity applications are those that allow users to connect to Internet (e.g. iPass), access their email (e.g.) UMI, access backed up files (e.g. Connected On-line).

In all these scenarios, user initially downloads the client from the JC platform and installs it on her desktop. User then launches the application by clicking on the corresponding link on her desktop. The application then prompts the user for login credentials. From a customer convenience viewpoint, the login credentials should be the same as JC platform. (. e.g. iPass, UMI Email).

In the case of Connected On-Line, if the user were to access her backed-up files over the web, she should be able to click on link through the portal, which should take her directly to the directory of files (without having to login to the application again).

9. Roles and Privileges

Refer Roles and Privileges FRD. Recall that the relationship amongst users, roles and privileges are n-to-n.

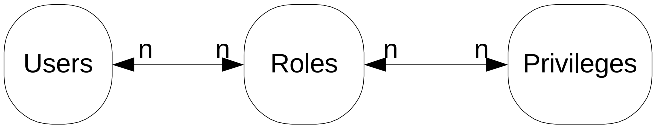


Figure 12: Users, Roles, & Privileges

For the sake of completeness, a short definition of roles and privileges is included-

A Privilege denotes availability of the "permission" to

- Take certain actions:

  Add a New User
  Search for a Company
  Create a New Workpage
  Edit a Content Module
  Create a New Role
  Delete a Company
  Provision Service for a User
  Activate a Service for a Company
  Request a New Service

- View certain Information Fields, Pages, Images, Buttons, Links, etc.:

  View User Profile Information
  View the Administrator Tab
  View the Upshot Service Icon
  View the Company Subscribed Services
  View the "Date Last Modified" Field
  View the User's Social Security Number Field
  See the "Update" Button on the Order Status Page
  View User's Salary Information

- Modify certain information:

  Change a User's Role (a specific field)
  Reset a User's Password
  Update a User's Home Address
  Modify the Order Status Field
  Modify the Service's Price for a Company
  Modify the Email Address Field
  Modify the Cost Center Code
  Modify the Location Type

Thus, a "Privilege" answers questions like –

- What can that User do?
- What can that User view?
- What actions can the User take?
- What fields can the User modify?

A Role is the name given to a particular set of Privileges.

We could associate each and every individual User in a system with a set of Privileges. However, Privileges are usually associated with a particular **job**, **title**, **position**, or **user group**.

For example, it is the "Human Resources Director" who has the authority to *View User's Salary Information*. The fact that John Doe is currently the Human Resources Director is what gives Mr. Doe the Privilege. When Mr. Doe leaves the organization, he will no longer continue to carry the Privileges associated with the Human Resources Director title.

Hence, the Privileges must be associated, not directly with the Users of any system, but with certain **categories** – which could be based on User's Division, Department, Job Title, User Group, etc. Such a logical set of "privileges" constitutes a Role.

9.1. Manage Privileges

The system should allow the ability to

- Create a specific privilege. (i.e.) Define a privilege name (or authorization policy) and a corresponding description of what the privilege is supposed to be.
- Modify/Update/Delete Privileges
- Classify information into intuitive groups. For example, we should be able to classify all Privileges related to adding and modifying Users into a "Manage User" Classification.
- Inherit one set of privileges to another.

9.2. Manage Roles

The security infrastructure should allow the ability to

- Create a Role
- Modify a Role
- Delete a Role
- Associate a set of Privileges to a Role
- Assign a Role to a User

10. Delegated Administration

As businesses expand, delegated administration makes it possible to delegate time consuming user administration workloads on various levels: within a Company (across departments, divisions, etc.); externally to Syndicators; and to individual users via a self-service function.

Delegated Administration comes in two flavors. As a way of

a. Separating functions.
b. Separating Homogenous resources.

10.1. Delegation By Separating Functions

1. Delegate the administration of User accounts, User Management and Company Management functions to the individual companies. This will free JC portal administrators or Syndicator administrators to focus on more important tasks such as defining security policies across the entire ecosystem.
2. Individual Company administrators can in turn, delegate these administrative functions to Department, Location, Regional Administrators, Help-Desk Managers, User's Managers and end-users themselves as appropriate.
3. Audit any delegated activities.
4. Delegate the administration of Companies within a specific Syndication model. For instance, when the Syndicator Administrator determines that he is getting swamped with managing the User Profile changes across all the Companies that are part of that Syndication. He could create an "Assistant Syndication Administrator" Child Role – without the "Delete" Privileges, but with the "User Profile Modification" Privileges, and then delegate this responsibility to another User from his Syndication Company.
5. Delegate the Billing and Provisioning Management activities within a Syndicator.

The following picture summarizes the various levels of delegation and what type of activities can be delegated at a Company level.

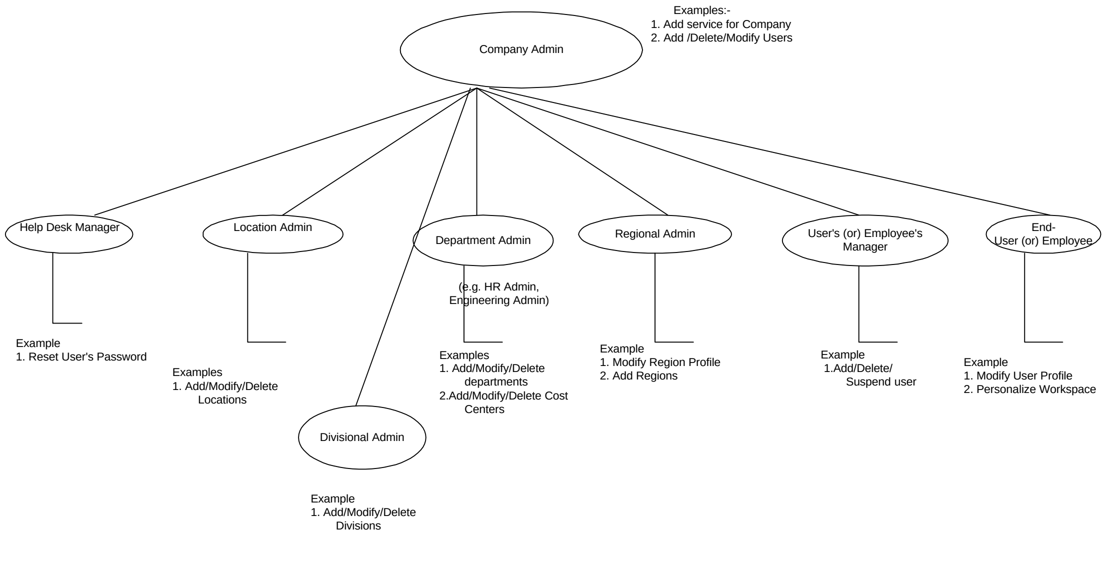
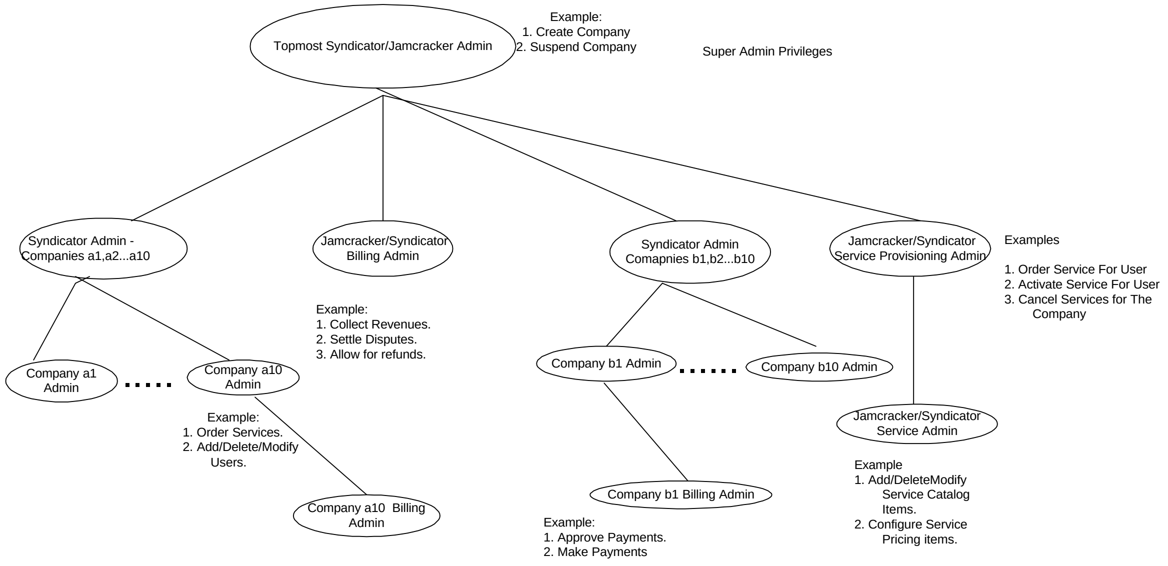


Figure 13: Delegated Administration Within Company – Scenario 1

The next diagram shows the type of activities that can be delegated with a Syndicator Admin.



1.

Figure 14: Delegation administration Within a Syndication (Jamcracker)

10.2. Delegation By Separating Populations of Homogenous Resources

In this scenario, delegation can be thought of as a way of separating populations of homogenous resources. For instance, Company Acme has a regional Admin in Boston who has rights to delete any one of users belonging to Boston, whereas Florida’s Regional Admin only has access to delete users belonging to Florida region. Thus, neither admin can delete the users in the others domain. The privileges for each admin are the same, namely “delete user”, but they apply to different sets of users.

- Add services to User -> "as"
- Delete services to User -> "ds"
- Disable Services to User -> "d"

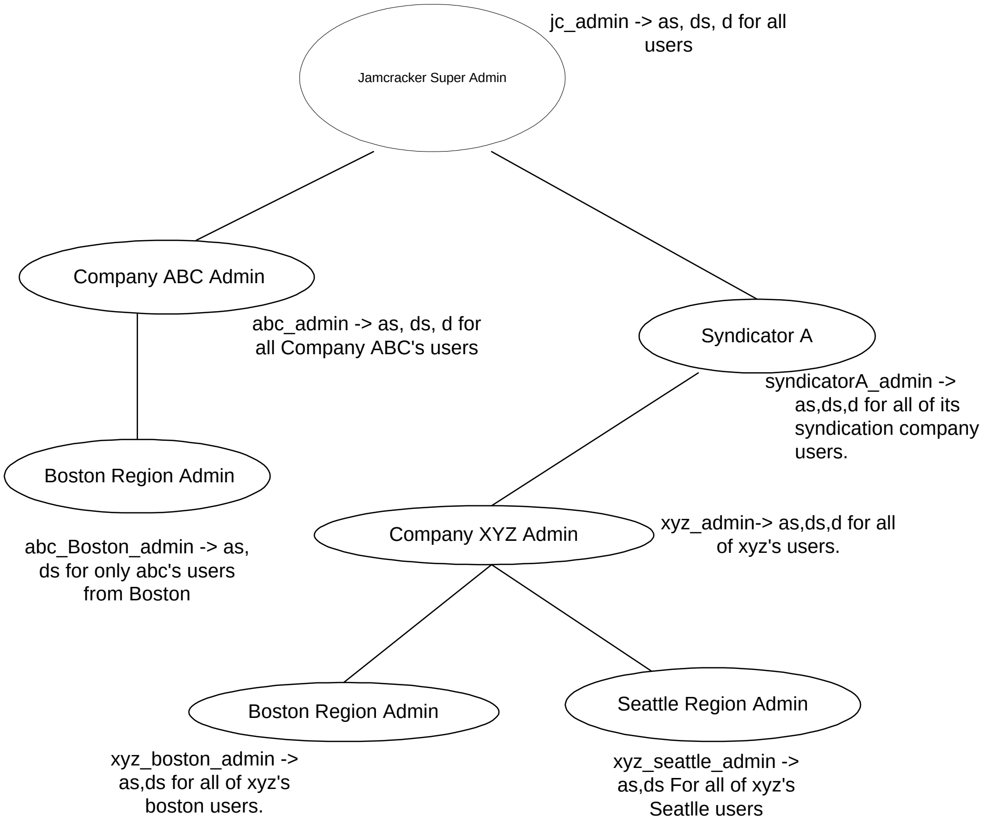

Jamcracker Super Admin
jc_admin -> as, ds, d for all users
Company ABC Admin
abc_admin -> as, ds, d for all Company ABC's users
Syndicator A
syndicatorA_admin -> as,ds,d for all of its syndication company users.
Boston Region Admin
abc_Boston_admin -> as, ds for only abc's users from Boston
Company XYZ Admin
xyz_admin-> as,ds,d for all of xyz's users.
Boston Region Admin
Seattle Region Admin
xyz_boston_admin -> as,ds for all of xyz's boston users.
xyz_seattle_admin -> as,ds For all of xyz's Seatlle users

Figure 15: Delegated administration – Scenario 2

As evidenced in the above diagram, even though Company XYZ’s Regional Admins have the same privileges, they operate on a different set of users.

10.3. Secure Delegation

All delegation activities must be administered securely. For instance, the Company must manage its own users and is prevented from seeing other users. Also, within the same Company, one department admin cannot perform the same functions on the users of a different department. Delegated administrators must be able to manage the roles and privileges as well as the users and security policies for which they have been granted explicit responsibility.

11. Session Management

To allow for seamless web experience across multiple sites, users sessions must be tracked across multiple web sites. Therefore, session management must be in place to capture and manage the users session from the time of initial login to final logout.

Certain applications like Financials comply with requirements such as SAS 70 certification, which require that the user's session or activity must be tracked or audited from login to logout. Therefore, from a platform perspective, a user's session must also be tracked to any of the ASP services that he or she may have accessed through the Workspace.

Thus, once user has logged on to JC platform and travel to various ASPs (or services), they should be able to do the following:

11.1. Source Time-out

If he or she remains idle on the platform for a specified period of time (which is configurable), the session on the platform should expire. If the sessions with other services are also inactive within the timeout period, then those must expire as well. This value should be configurable per Company. The system must pop up a timeout reminder at least 10 minutes before the session times out.

11.2. Destination Time-out

So long as the user has an active session on the JC platform, he or she should remain active on the destination applications participating in SSO as well. However, each application (based on the sensitivity – e.g. financial application) should be given the choice to implement a session time-out that would either override or comply with the time-out parameter of the JC platform.

Timeout values should be configurable on a per Company basis. JC should set a liberal maximum value. Each ASP could set their own maximum value and each company could set their own maximum value. The "effective" time-out value should be the minimum of these values.

11.3. Global Logout

If he or she logs out of the platform, they should be logged out of the all the services participating in SSO as well.

11.4. User Lockout

If the user were to be locked out for some reason, say browser crashes, the period that he or she has to wait before logging in again should be configurable per Company. This parameter must be separately configured from session timeout parameter.

11.5. Concurrent Logins

The system must allow for unlimited number of multiple concurrent sessions.

The system must track these sessions so long as the user's session is active within the Jamcracker Platform.

11.5.1. Multiple Concurrent Logins

Each user should be allowed to open up multiple concurrent sessions with the JC platform.

When users logout of their session at the JC platform by clicking on the logout button, any session information must be cleared from a user's browser cache so that one user cannot access another user's data from a within the same session.

Similarly, if the user were to logout by closing the browser window, he or she should be able to log into the portal after a session time-out period is applied. This session time-out should ideally be configurable per Company.

11.5.2. Session Timeout Parameter

If the user remains idle on the platform for a specified period of time (which is configurable), the session on the platform should expire. This value should be ideally configurable per Company (or) set to a pre-defined limit at a session level. The system must pop up a timeout reminder at least 10 minutes before the session times out.

11.5.3. Logging Requirements

System must log the following information whenever a user opens and closes up multiple sessions within the Jamcracker platform

- Company name, userid, session#, a description of the event i.e.. no. of multiple sessions, time, date
- Session closeout details - Either timeout or user logout for each session that was open.

11.6. Cookie Management

Jamcracker currently uses a permanent cookie (created on user's desktop) to store the Company acronym name for logging into Jamcracker platform. The cookie does not contain any personal information and is used mainly for pre-populating the login screen with the Company acronym whenever the user logs in.

Although this is convenient for a user, this is unacceptable for companies that are security conscious. This is because permanent cookies leave crumbs or trail from which a user's visit to Jamcracker could be traced – poses an invasion of privacy issue for many users.

Hence the system must not use permanent cookies.

12. Auditing and Logging

The security infrastructure of any service provider should support auditing so that unusual user activity can be identified quickly and the offending user dealt with immediately. Digital paper trails of sensitive applications (like Financials) must be available

12.1. Security Events for Logging & Auditing

Security infrastructure should allow any event that is of interest to the Company be logged and subsequently audited. Typically any administrator level activity and any user activity on the JC workspace must be tracked (i.e. logged). It should provide the ability to

1. Log any administrative changes. (For example, any change to admin roles)
2. Track online any administrative activity, available for future auditing.
3. Log resource usage.
4. Support synchronous event logging – (i.e.) Operation will fail if the event cannot be logged.
5. Allow for all logging activities to happen within a specific time-period.
6. Any password resets.
7. Password modifications
8. Change Username
9. Change Company Name
10. Add/Modify/Suspend/Delete Company
11. Add/Modify/Suspend/Delete User
12. Add/Modify/Suspend/Delete Service for Company
13. Add/Modify/Suspend/Delete Service for User
14. Modify User Role
15. Log multiple concurrent sessions (both open and closeouts)– Company name, userid, session#, a description of the event i.e.. No. Of multiple sessions, time, date
16. Log multiple session closeout details - Either timeout or user logout for each session that was open.
17. Log the following details whenever a user's account is disabled - User's full name, Login name, Company name, Date, Time of disabling, name of the Company Admin who disabled the account
18. Log the following details whenever a user's account is re-enabled - User's full name, Login name, Company name, Date, Time of re-enabling, name of the Company Admin who re-enabled the account
19. Log the following details whenever a user's account is locked
    User's Full Name, Login Name, Company Name, Date, Time, Originating IP, Reason for locking (e.g. invalid password or invalid username)
20. Log the following details whenever a user's account is unlocked.
    User's Full Name, Login Name, Company Name, Date, Time, Name of the Admin (JSC or Syndicator or Company Admin) who unlocked the account

For instance, a Company may request JC to provide an audited report of all activities within the last 1 week for an ex-employee who had the role of a Company Administrator.

Another example, Company may want online tracking (or auditing) of a Financial Administrator using Oracle Financial Apps. In both these scenarios, JC must be able to provide those reports.

12.2. Logged Reports Available through the JC Workspace

Audited/logged reports must be available through the JC workspace for authorized users. Special privileges to view audited report must be available.

13. Meta Directory Synchronization

This section discusses the security requirements during the process of Meta directory (e.g.) LDAP synchronization. JC platform allows the capability to perform a LDAP to LDAP synchronization between Customer and JC. Since the data flow involves customer sensitive information over to JC (and ASP), the following security requirements must be met.

13.1. Data Privacy

Privacy of the data must be maintained during the synchronization process. That is, the communication between JC and Customer, JC and ASP must be secure during the synchronization process.

13.2. Integrity

Integrity of data must be preserved during and after the transaction. See section on encryption that can help achieve this requirement.

13.3. Non-Repudiation

Non-repudiation must be guaranteed during and after the transaction.

13.4. Mutual Authentication

Mutual authentication should take place between the two parties exchanging data, for example between customer and JC (or) JC and an ASP.

13.5. Authorization

Only authorized users should be given access to update the master LDAP. (A special role with proper privileges to update/create/delete entries in JC LDAP should be created to perform the synchronization.)

13.6. Customer Data Segregation

If the Service Provider (in this case, either JC or the ASP) combines the data of different customers on shared physical servers, there should be a documented set of controls that will ensure separation of data and sensitive information between customers.

13.7. Audit Trail Availability

An audit trail of the synchronization activity must be available if requested by the customer.

13.8. Customer and JC LDAP synchronization

The following describes the various scenarios of LDAP synchronization between Customer and JC. In all scenarios, authentication information consists of valid username/password combination as well as access information to LDAP source tree.

13.8.1. Pull data from customer LDAP data source

Here, JC is responsible for maintenance of the data replication process as well as the accuracy and integrity of the data.

The data is pulled from the customer LDAP at predefined time intervals agreed upon by the customer and JC. This process replicates data from the customer LDAP and populates the JC master LDAP. Data replication is incremental and one-way (from the customer to JC).

13.8.2. Get data push from customer LDAP data source

The customer is responsible for maintenance of the data replication process and the accuracy and the integrity of the data.

The data is pushed from the customer LDAP at predefined time intervals agreed upon by the customer and JC. This process replicates data from the customer LDAP and populates the JC master LDAP. Data replication is incremental and one-way (from the customer to JC). The customer chooses the method/tools that is compatible with the JC LDAP server.

13.8.3. Customer pulls data and populates/updates their LDAP from JC Master LDAP

The customer is responsible for maintenance of the data replication process and the accuracy and the integrity of the data.

The data is pulled to the customer LDAP at predefined time intervals agreed upon by the customer and JC. This process replicates data from the JC master LDAP and populates the customer LDAP. Data replication is incremental and one-way (from JC to customer). The customer chooses the method/tools that is compatible with the JC master LDAP server.

13.8.4. JC Master LDAP pushes data to populate and update customer LDAP

Jamcracker is responsible for maintenance of the data replication process and the accuracy and the integrity of the data.

JC master LDAP pushes data to customer LDAP at agreed upon predefined time intervals. This process replicates data from the JC master LDAP and populates the customer LDAP. Data replication is incremental and one-way (from Jamcracker to Customer)

13.8.5. Pull Data from Customer w/o storing internally

Jamcracker also locally stores the userid and provisioning information locally for billing purposes and internal verification. No password or authentication information will be stored locally. The customer is responsible for maintenance of the LDAP, the accuracy and the integrity of the data.

The data is pulled from the customer LDAP at user login. The userid existence for the customer is authenticated against the customer LDAP. The service provisioning is authenticated with JC data sources. During the user authentication the user service provisioning information is retrieved and compared against the local data for the user. The local user information is updated (the need for maintaining history is important for billing reasons) if the provisioning information does not match.

13.8.6. Hosted LDAP service to customers

Customer incorporates the LDAP schema provided by JC into their LDAP schema. Customization to the schema is done beyond the JC LDAP schema. The customer has access to add and modify user profile information. Customer is responsible for maintenance of the data and the accuracy and the integrity of the data. The LDAP server is hosted at and administered by Jamcracker.

13.9. JC and ASP LDAP synchronization

(To do: - Will expand this section later.)

14. Password Policies

JC must enforce a set of minimum standards as not to expose other companies to attacks that could prove lethal to both JC and other Companies. Thus, having a standard password policy that comprises the following functionality will help achieve that goal.

14.1. Reset Passwords And Retrieve Usernames

There are 2 situations in which Reset Password and Retrieve Username is applicable:
User enters incorrect password to Jamcracker Central.
User enters incorrect login credentials to Jamcracker Central.

In both cases, he or she can get help to reset their password or retrieve their User name from Jamcracker Central via the “Login Help” Screen.

14.1.1. Reset Password Choices – New reset values

Whenever the password is reset, the new value is one of the following: -

- First Time Login Password - Single Password for Entire Company
- First Time Login Password – Known Unique password for Each User
- Any arbitrary value as desired by Company Admin

The Company Admin must be given a choice of any of the above three during the process of resetting the user’s password. Whereas JSC Admin or Syndicator Admin will only have the second choice.

14.1.2. Reset Password Performed By The User Directly

In this instance, user has access to Jamcracker central and is typically connected to the Internet via ISP (third-party provider or Company LAN).

To reset the User’s password the following data fields are required:
Company Name – Users Company Name
Login Name – Users Login Name
Email Address – Users email address.
Secret Question – Choices from A through F
Secret Answer –Answer to secret question.

Recall that the user selects the Secret question and Secret answer when he or she logs into Jamcracker central for the first time.

The end User then selects, via a click, to submit or Clear the form.

If the user successfully enters all the above, an acknowledgement screen is displayed. The system immediately logs the user in and prompts for a new password (Note – The system will not prompt for the old password since the user does not know this.).

An email acknowledging the password reset event should be sent to the user. The email should not contain any information about the new password. The email should indicate the date/time, requestor and the fact that the password has been successfully changed.

If unsuccessful, the system displays a “We are sorry…” screen. If the User fails 3 times the User’s account will be locked for TBD minutes.

An email should be sent to the Company Admin and the user acknowledging the failed attempt. The email should indicate that if the user did not request the password change, they should contact the Jamcracker Service Center (Syndicator Admin) immediately.

The system should log the unsuccessful attempt for future auditing.

14.1.3. Reset Password Performed by Jamcracker or Syndicator Admin (JSC)

In this instance, user has no access to Jamcracker central. This is applicable in cases where user's ISP (e.g. iPass) is tied to the Jamcracker central login credentials. User calls the JSC or their Syndicator help-desk and asks to reset the password.

To reset the User's password the following data fields are required:

Company Name – (Users Company Name)
Login Name – Users Login Name
Email Address – Users email address.
Secret Question – Choices from A through F
Secret Answer –Answer to secret question.

JSC (or Syndicator) representative asks the user the answers for all of the above questions over the phone.

If user answers successfully, the password is reset to the second choice presented in 14.6.1. JSC/Syndicator rep can give the password over the phone to the user.

An email acknowledging the password reset event should be sent to the user as well as the Company Admin. The email should not contain any information about the new password. The email should indicate the date/time, requestor and the fact that the password has been successfully changed

If not successful, JSC/Syndicator representative should convey that over the phone. If the attempt fails more than once, JSC Admin/Syndicator Admin should ask the user to contact their Company Admin. Support ticket should be escalated to the Company Admin.

An email should be sent to the Company Admin as well as the user acknowledging the failed attempt. The email should indicate that if the user did not request the password change, they should contact the Jamcracker Service Center (Syndicator Admin) immediately.

The system should log the unsuccessful attempt for future auditing.

14.1.4. Reset Password Performed By Company Admin

In this instance, the user calls up their Company Admin directly. This is applicable in scenarios where user has forgotten their secret question or answer.

To reset the User's password the following data fields are required:

Company Name – Users Company Name
Login Name – Users Login Name
Email Address – Users email address.
Secret Question – Choices from A through F
Secret Answer –Answer to secret question.

The Company Admin has a choice whether or not to ask these questions over the phone to the user.

Company admin can directly go to the Admin interface through the Jamcracker and reset the user's password. The new reset value is either of the choices presented in 14.6.1. Company Admin can give the password over the phone to their end-user.

An email acknowledging the password reset event should be sent to the user as well as the Company Admin. The email should not contain any information about the new password. The email should indicate the date/time, requestor and the fact that the password has been successfully changed

14.1.5. Retrieve Username Performed By User

In this instance, user has access to Jamcracker central and is typically connected to the Internet via ISP (third-party provider or Company LAN).

To retrieve the User's login name the following data fields are required:
Company Name – Users Company Name
Zip Code – Zip code of Users Company
Email Address – Users email address.
Secret Question – Choices from A through F
Secret Answer –Answer to secret question

The end User then selects, via a click, to submit or Clear the form.

If the user successfully enters all the above, an acknowledgement screen is displayed. The system immediately logs the user in.

An email acknowledging the Retrieve Username event should be sent to the user. The email should contain the user's login name as well.

If unsuccessful, the system displays a "We are sorry…" screen. If the User fails 3 times the User's account will be locked for TBD minutes.

An email should be sent to the Company Admin as well as the user acknowledging the failed attempt. The email should indicate that if the user did not request the login retrieval, they should contact the Jamcracker Service Center (Syndicator) immediately. The system should log unsuccessful attempts for future auditing.

14.1.6. Retrieve Username Performed By Jamcracker or Syndicator Admin (JSC)

In this instance, user has no access to Jamcracker central. This is applicable in cases where user's ISP (e.g. iPass) is tied to the Jamcracker central login credentials. User calls the JSC or their Syndicator help-desk and asks to retrieve the username.

To retrieve the User's login name the following data fields are required:

Company Name – Users Company Name
Zip Code – Zip code of Users Company
Email Address – Users email address.
Secret Question – Choices from A through F
Secret Answer –Answer to secret question

JSC (or Syndicator) representative asks the user the answers for all of the above questions through the phone.

If user answers successfully, JSC/Syndicator rep can give the username over the phone to the user. An email acknowledging the retrieve username event should be sent to the user. The email should contain the new username.

If the attempt fails more than once, the User's account will be locked for TBD minutes. The JSC/Syndicator representative should convey that over the phone the user and ask the user to contact their Company admin. Support ticket should be escalated to their Company Admin as well.

An email should be sent to the Company Admin as well as the user acknowledging the failed attempt. The email should indicate that if the user did not request the login retrieval, they should contact the Jamcracker Service Center (Syndicator) immediately. The system should log unsuccessful attempts for future auditing.

The system should log unsuccessful events for future auditing.

14.1.7. Retrieve Username Performed By Company Admin

In this instance, the user calls up their Company Admin directly. This is applicable in scenarios where user has forgotten their secret question or answer.

To retrieve the User's login name the following data fields are required:

Company Name – Users Company Name
Zip Code – Company's zip code
Email Address – Users email address.
Secret Question – Choices from A through F
Secret Answer –Answer to secret question.

The Company Admin has a choice whether or not to ask these questions over the phone to the user.

Company admin can directly go to the Admin interface through the Jamcracker and retrieve the user's login name. Company Admin can give the username over the phone to their end-user.

An email acknowledging the retrieve username event should be sent to the user. The email should contain the new username.

14.2. Challenge Questions For Password Authentication

JC should provide a list of challenge questions from a known list of data fields such as

a. What is your Company Name
b. What is your Company Zip Code
c. What is your email address
d. What is your secret question?

from which a Company can select what their challenge questions will be.

14.3. Challenge Phrase For Password Authentication

User will typically respond with a challenge phrase. System should verify the answers and the challenge phrase and once validation is done, the password will be reset.

14.4. Password Expiration

Every Password should have an expiration time associated with it. JC/Syndicator should provide a list of choices within which any Company can choose their own password expiration time. (e.g.) None, 30, 45, 60, 90 days.

14.5. First Time Login Password

There are 2 situations in which First Time Login in applicable:

- The First Time **ever** a user logs into the system
- The First Time a user logs into the system after his password is **reset** by an Admin (only 14.5.1. and 14.5.2)

The type of First Time Login methodology used must be configurable for each company. In some instances, they may want a highly secure mechanism, which would involve significant work for the CompanyAdmin to collect the password and/or communicate it to his users. In others, they may want to reduce the administrative workload on the CompanyAdmin and have a less stringent security requirement.

14.5.1. First Time Login Using a Single Password for Entire Company

In this instance, the same password is used for first time login for all users in the company. The primary goal is to reduce the administrative workload on the CompanyAdmin. This is inherently less secure since every user knows your first time login password.

Eamples include:

- “pass4Uword”
- “jam4u!”
- “rock3!et”
- “foobar123”
- JAM060101” - The word “JAM” appended to the company’s deployment date. For example, if the company went alive on 06/11/01, then the first time password would be “JAM061101”.

The key requirement for the above examples is that it should **NOT** be defined in any dictionary.

14.5.2. First Time Login Using a Known Unique Password for Each User

In this instance, each user has a unique password. The user can determine this password – because he knows the answer to the question. The Company Admin is responsible for providing this password to Jamcracker.

The goal is to be secure **and** reduce administrative workloads.

The key requirement is that the field should not be viewable through the user directory.

Examples include:

- “username”
- “birthdate”
- “employee number”
- “mothers maiden name”
- “hiredate” etc.

14.5.3. First Time Login Using a Random Unique Password for Each User

In this instance, each user has a unique password. This password is randomly generated by the Company Admin (or Jamcracker) for each user. The Company Admin is then responsible for communicating each random password to each respective user. If the Company Admin generated these passwords, then he is also responsible for providing these passwords to Jamcracker.

Either case whoever generates the random passwords (Company or Jamcracker) must send these in an encrypted fashion to the other party. For example, if Jamcracker generates random passwords for a Company, Jamcracker must send the passwords as an encrypted (self-decrypting) email attachment to the Company Admin.

Since these are randomly generated, it is likely that users will **forget** these passwords. So, the Company Admin will need to be able to provide these to users in the future.

Examples include:

- Random Unique Password for Each User generated by Company Admin
- Random Unique Password for Each User generated by Jamcracker

14.5.4. First Time Login Using a Digital Certificate for Each User

In this instance, each user will have a unique PKI-based digital certificate. This represents the highest level of security but is very **complex** to implement. A 3rd party PKI provider will be responsible for providing the PKI infrastructure. This is unlikely to used in the near term.

14.6. Change Password

Users should be allowed to change their password at any point in time. Password change must allow for a delay of a certain minimum number of days to ensure users do not change their password ‘n’ times in order to reset it back to their expired password. For instance, the system must enforce a password rule that limits the number of password changes that a user may make in any given day.

14.7. Password History

A password history must be maintained so that users do not choose the same password over and over again. However, the number of entries in the password history must be Company configurable. It must be set no lower than a floor value, say between 4 and 8.

14.8. Password Syntax

Companies should be allowed to choose the syntax for their passwords. For instance, the password should contain a combination of 12 characters of which 4 should be numbers etc.

14.9. Password Management Between JC and ASP

JC and the ASP should have a commonly agreed upon format for password exchange during Simple Single Sign-On. For instance, every user’s password that is exchanged during SSO between JC and ASP must be changed every 90 days, encrypted and have a minimum length of 8 characters.

14.10. Last Login

System should display the last login information for the user. The information should include date and time when last logged in.

14.11. Minimum And Maximum Length

Every password should have an acceptable minimum length and maximum length associated with it. JC/Syndicator should set the limits on the minimum and maximum length (example, 6 characters to 40 characters.) so that every Company can choose a minimum and maximum within this pre-determined limit.

14.12. Account Locking

User accounts must be locked after specified number of invalid attempts to login into the JC platform. However, administrator accounts should not be subject to the same restrictions.

A user's account can be locked under the following situation:

- User fails to successfully login to the portal after 3 consecutive attempts.

14.12.1. User Account Locking

The system should clearly state if the invalid login attempt was due to any of the following:

- Invalid password
- Invalid Username

No one except the user can lock himself or herself out of the portal.

[Note: - A separate field called **lockstatus** should be created – it merely states whether the user's account is locked or unlocked.]

The **lockstatus** of the user's account must change to "LOCKED".

The message presented to the user upon lock out should be something like
"Your User name or password is incorrect and your account is now in a locked
state. Please contact the Jamcracker Service Center (or Syndicator Admin)
for additional assistance

Once the user's account is locked, he or she can "unlock" the account by doing the following:

14.12.2. User account unlocking – Performed by JSC admin or Syndicator Admin

In this case, user calls up JSC Admin or their Syndicator Admin to unlock the account. JSC Admin or Syndicator Admin must go through a password-reset scenario as described in 14.6.3.

The lockstatus field of the user's account must change to "UNLOCKED".

14.12.3. User account unlocking – Performed by Company Admin

In this case, user calls up Company Admin to unlock their account. Company Admin must go through a password reset scenario as described in 14.6.4.

The lockstatus field of the user's account must change to "UNLOCKED".

14.12.4. Logging Requirements for Account Locking and Unlocking

The system should log the following details whenever a user's account is locked

a) User's Full Name, Login Name, Company Name, Date, Time, Originating IP, Reason for locking (e.g. invalid password or invalid username)

All logged items should be available for future auditing.

The following details must be logged whenever a user's account is unlocked.

a) User’s Full Name, Login Name, Company Name, Date, Time, Name of the Admin (JSC or Syndicator or Company Admin) who unlocked the account

14.13. Account Disabling

[Note: - This functionality is already described under “User Management FRD” – See section under Suspend/Enable a User.]

The following situations may cause a user’s account to be disabled.

- Company Admin can disable a user’s account.
- JSC Admin can disable a user’s account.

No one except the Company Admin or JSC/Syndicator Admin can re-enable the user’s account.

14.13.1. Company Admin Performs User Account Disabling

A Company Admin can disable or suspend a user’s account for several reasons such as

- User is no longer with the Company
- User went on long leave on absence etc.

From a security standpoint, access to the user’s account is temporarily suspended.

An email should be sent to JSC Admin or Syndicator Admin containing the following: -

a. Name of the Company Admin who disabled the account
b. User’s full name, login name
c. Date and time of disabling the account

The status of the user’s account must change to “SUSPENDED”. The system should log the account-disabling event for future auditing.

14.13.2. JSC Admin or Syndicator Admin Performs User Account Disabling

A JSC Admin or Syndicator Admin performs user account disabling based on a Support request either from the Company Admin (or) user’s Manager.

An email should be sent to the Company Admin containing the following: -

d. Support Tracking number
e. User’s full name, login name
f. Date and time of disabling the account

The status of the user’s account must change to “SUSPENDED”.

14.13.3. User Account Re-enabling – Performed By Company Admin

Both the Company Admin and the JSC/Syndicator Admin can re-enable the user’s account. The system must provide an UI for the Company Admin to re-enable the account.

An email should be sent to Jamcracker or Syndicator Billing Admin with the following

g. Name of the Company Admin who initiated the re-enabling
h. User’s full name, login name
i. Date and time of re-enabling the account

The status of the user’s account must change to “ACTIVE”.

14.13.4. User Account Re-enabling – Performed By JSC Admin or Syndicator Admin

A JSC Admin or Syndicator Admin can re-enable the user's account based on a Support request either from the Company Admin.

An email should be sent to the Company Admin containing the following: -

j. Support Tracking number
k. User's full name, login name
l. Name of the Company Admin who initiated the re-enabling
m. Date and time of re-enabling the account

The status of the user's account must change to "ACTIVE". The system should log the account re-enabling event for future auditing.

14.13.5. Logging Requirements for Account Disabling And Re-Enabling

The system should log the following details whenever a user's account is disabled.

1. User's full name, Login name, Company name, Date, Time of disabling, name of the Company Admin who disabled the account

The system should log the following details whenever a user's account is re-enabled.

2. User's full name, Login name, Company name, Date, Time of re-enabling, name of the Company Admin who re-enabled the account

14.14. Login Retries

Companies must be allowed to choose the number of attempts (or retries) after which their user accounts will be locked. For instance, some companies may want their user accounts to be locked out after 4 retries, whereas some may prefer 6. JC must set a maximum default value so that Companies can choose a value that is less than the maximum.

15. Syndicator Security Requirements

With regard to security requirements, a syndicator Partner should be able to do the following

15.1. Authentication

Define the authentication policies for their direct chain of customers. For instance, if Syndicator A manages Companies A1, A2, A3 and A4, Company A1, A2 and A3 may use basic authentication whereas A4 may use strong form of authentication.

15.2. Authorization

Syndicators must be able to define what roles and privileges the users of the direct chain of companies must have. They should be able to define their own policy domains.

15.3. Encryption

Specify what data fields should be encrypted, type and strength.

15.4. Secure SSO

Specify what types of applications should be integrated through SSO from JC workspace as well what form of SSO should be used. Customers of syndicators should have Secure SSO from the JC workspace into non-affiliated ASPs.

15.5. Delegated Administration

A Syndicator Admin must be able to manage all the Roles and Privileges for all company's users as well as delegate certain administrative activities to the Company Admins.

15.6. Session Management

Support global logout, source and destination time-outs.

15.7. Auditing & Logging

Specify what events should be audited and what activities should be logged for their direct chain of companies.

15.8. Meta Directory Synchronization

Satisfy the security requirements of Data privacy, integrity, non-repudiation, mutual authentication, authorization, customer data segregation and audit trail availability.

15.9. Password policies

1. Specify the default for password expiration time
2. Specify the default minimum length
3. Specify the default maximum length
4. Specify a valid list for number of password retries

16. Company Security Requirements

The following section discusses the Company Security Requirements.

16.1. Authentication

Companies should be able to choose what form of authentication they need, whether basic or strong.

16.2. Secure SSO

Any valid user of the Company should be able to do a Secure Single Sign-On to various types of applications that the platform integrates to.

16.3. Roles and Privileges

Companies should be able to create and configure their own Roles with their own specific Role Names as well as define their own policy domains.

16.4. Delegated Administration

Company Admin should be able to delegate management of roles to an assistant Company Administrator. For instance, a Company Admin may decide to delegate all user administration of all U.S. employees to U.S. Company Admin whereas Asia-Pacific Admin will administer Asia-Pacific employees.

16.5. Auditing & Logging

Companies should have the ability to define what events need to be logged as well as the data that needs to be encrypted. As mentioned earlier, all logging activities should have a time-period associated with it. For instance, a Company may specify that all the activities of their Company Admin must be tracked from January 1 through 15th. Select from a list of pre-defined events that can be audited and select what activities need to be logged.

16.6. Meta Directory Synchronization

If companies were to choose Directory Synchronization, then security requirements specified in Section 13 should be satisfied.

16.7. Password policies

Company Admin must be able to configure the following:

1. Select the password expiration time.
2. Select the minimum length.
3. Select the maximum length.
4. Select Number of Password retries
5. Specify Password syntax.
6. Specify first-time Login password – Company Wide (or) Specify first time Login Password convention for each user.
7. Enable or Disable Password History
8. Select No. Of entries for Password History
9. Select challenge phrase questions from a list of allowable ones.

17. User Security Requirements

From an end-user perspective, all communications with JC and its ASP's must be secure. Users should be able to

17.1. Authentication

Authenticate him or her either with basic authentication or strong authentication (depending on the Company's authentication policy) with JC workspace.

17.2. Secure SSO

Perform a secure SSO to all of the applications that the platform integrates with.

17.3. Roles and Privileges

See personalized content on JC as well as various applications based on their roles and privileges.

17.4. Session Management

1. Log out of JC workspace and simultaneously be logged out of other ASP sessions as well.
2. Login to JC workspace from work, home, library, public terminals (at airport etc).
3. Use any device including PC, PDA, and Wireless etc.

Finally, user-experience to JC workspace must be seamless.

User should be able to configure the following items.

17.5. Password Policies

1. Select challenge phrase for password authentication.
2. Change Password.
3. Reset Password.

18. ASP Security Requirements

From a security perspective, ASP should be able to do the following

18.1. Authentication

Whatever form of authentication JC supports, ASP should trust JC and participate in SSO.

18.2. Authorization

Support a centralized authorization model and leverage authorization policies from JC.

18.3. Encryption

Encrypt all customer sensitive data and all sensitive customer communication.

18.4. Session Management

Support all session management functionality such as being able to track users sessions, support destination time-outs and global logout functionality.

18.5. Auditing & Logging

Any events specified by Company or Syndicator should be logged. Logged reports must be made available to JC.

18.6. Meta Directory Synchronization

Satisfy all the security requirements required for Directory Synchronization specified in Section 13.

18.7. Password Policies

Password Management for SSO – Change password every ‘N’ days where ‘N’ is configurable between JC and the ASP.

19. Security Requirements For Outsourced Help-Desk Partners

Jamcracker has business requirements to outsource part of its help-desk operations to 3rd parties located both domestically here in the US as well as internationally. The outsourced partners have many help-desk agents who will be taking the first support call from the customer.

Will need to revisit this section on security requirements such as
a). Do we want to allow admin level privileges to outsourced partners?
B) What level of access should they be given? Maybe just reset passwords, manage data fields for applications other than HR.

19.1. Outsourced CSR Role

For instance, a separate role could be created for outsourced partners so they

1. Cannot reset passwords.
2. Cannot Delete services
3. Cannot Add Services (usage based).

20. Changing User Names and Company Names

System must allow for the following functionality:

20.1. Change Username

Users must be allowed the flexibility to change their names as well as their login ids. For instance, Mary Jones could marry and become Mary Smith. The login name must be changed from mjones to msmith. The system should therefore be flexible enough so as to allow for this without breaking any of the functionality such as Secure Single Sign-On.

20.2. Change Company name

Companies should be allowed to change their login names as well. For example, Company ABC merges with Company XYZ and the new Company is called DEF. Therefore, system should allow ABC to change its acronym from ABC to DEF.

21. Configurable Items

This section covers attributes, which should be configurable through a GUI (integrated into the workspace).

21.1. Password Expiration Time

Password Expiration – How often should the password expire? Expiration time varies from one company to another (e.g. 90 days, 60 days etc.).

21.2. Maximum Password Length

Maximum Length – How long should the password be? JC or Syndicator should specify a pre-determined maximum length. Companies can then set their own maximum length within this pre-determined limit.

21.3. Minimum Password Length

Minimum length –This specifies what the minimum length of the password should be. JC or Syndicator should specify a pre-determined minimum length. Companies can then set their own minimum length within this pre-determined limit.

21.4. No. Of Password Retries

Number of times a user can retry his or her login credentials after which the account should be locked. This is applicable if the account-locking feature is enabled. Companies can then choose from a valid list of choices.

21.5. First-Time Login Password - Company

Initial password when a Company is first provisioned at JC Workspace. This would be applicable to all users within the Company. Alternately, Companies can specify a convention for first time password for each user (for instance, first time password will be the same as login id). The system should prompt for a change of password as soon as first time users log in.

21.6. Password History – Enable or Disable

Password History – If enabled, this would allow the system to maintain a history of passwords. Whenever, the user changes his or her password, system must check to make sure that it is not stored in the history. Companies must be allowed to enable or disable this feature.

21.7. Password History – No. Of Entries

This attribute specifies the number of entries that the password history file must have. It should be configurable at a Company level.

21.8. Challenge Phrase for Password Authentication

This attribute is configurable at a user level and would specify what secret answer or phrase must be used for password validation in case user has forgotten his or her password. The challenge question is pre-determined by JC from a list of allowable questions.

21.9. Challenge Questions for Password Authentication

JC/Syndicator should provide a list of challenge questions from a known list of data fields such as

a. What is your Company Name

b. What is your Company Zip Code
c. What is your email address
d. What is your secret question?

Companies must choose from this allowable list.

21.10. Password Syntax

Companies should be allowed to specify the syntax of their passwords. Syntax should consist of a combination of letters, numbers and special characters.

21.11. Events For Logging

Here are some examples of events, which could be logged. Some of the events should also be logged all the way up to the ASP (Delete a Company, Add Company, Delete User, Activate Service for Company etc.)

- Add/Modify/Suspend/Delete Company
- Add/Modify/Suspend/Delete User
- Add/Modify/Suspend/Delete Service for Company
- Add/Modify/Suspend/Delete Service for User
- Modify User Role
- Modify Admin Role
- Reset/Modify Password
- Change User Name
- Change Company Name
- Modify User Profile
- Unauthorized login attempts
- Log multiple concurrent sessions (both open and closeouts)– Company name, userid, session#, a description of the event i.e.. No. Of multiple sessions, time, date
- Log multiple session closeout details - Either timeout or user logout for each session that was open.
- Log the following details whenever a user's account is disabled - User's full name, Login name, Company name, Date, Time of disabling, name of the Company Admin who disabled the account
- Log the following details whenever a user's account is re-enabled - User's full name, Login name, Company name, Date, Time of re-enabling, name of the Company Admin who re-enabled the account
- Log the following details whenever a user's account is locked -User's Full Name, Login Name, Company Name, Date, Time, Originating IP, Reason for locking (e.g. invalid password or invalid username)
- Log the following details whenever a user's account is unlocked. User's Full Name, Login Name, Company Name, Date, Time, Name of the Admin (JSC or Syndicator or Company Admin) who unlocked the account

Basic logging should help us when we are audited and need to prove who did what. So, here is the basic information that needs to be captured:

- Event Timestamp
- Event performed by UserName/CompanyName (sometimes these will be Jamcracker Employees)
- Event performed on UserName/CompanyName (sometimes these will be Jamcracker Employees)

22. Wireless Security

(To be expanded later….)